Einführung in Visual Computing

186.822

Lighting and Shading

Werner Purgathofer

Shading in the Rendering Pipeline

object capture/creation

scene objects in object space

modeling

viewing

projection

vertex stage („vertex shader“)

transformed vertices in clip space

clipping + homogenization

scene in normalized device coordinates

viewport transformation

rasterization

shading

pixel stage („fragment shader“)

raster image in pixel coordinates

light sources

basic model

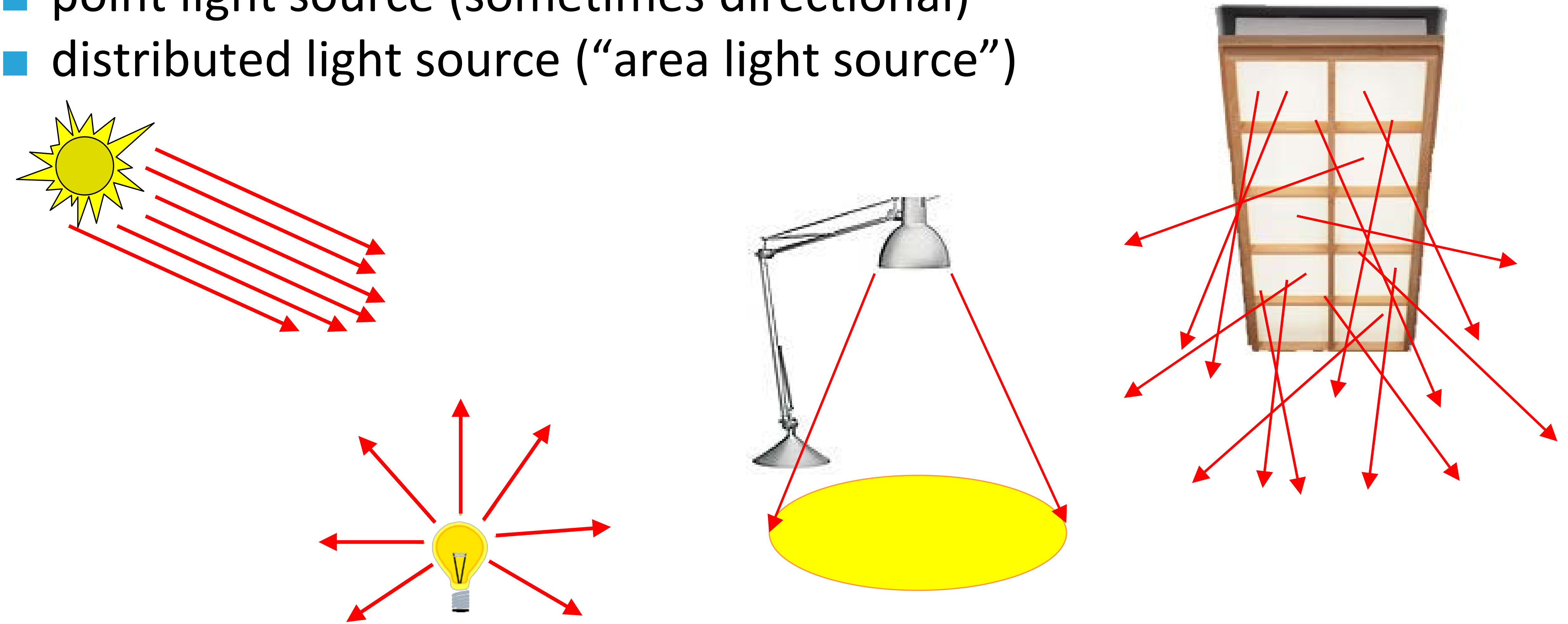
- ambient light
- diffuse shading
- specular highlights

shading interpolation

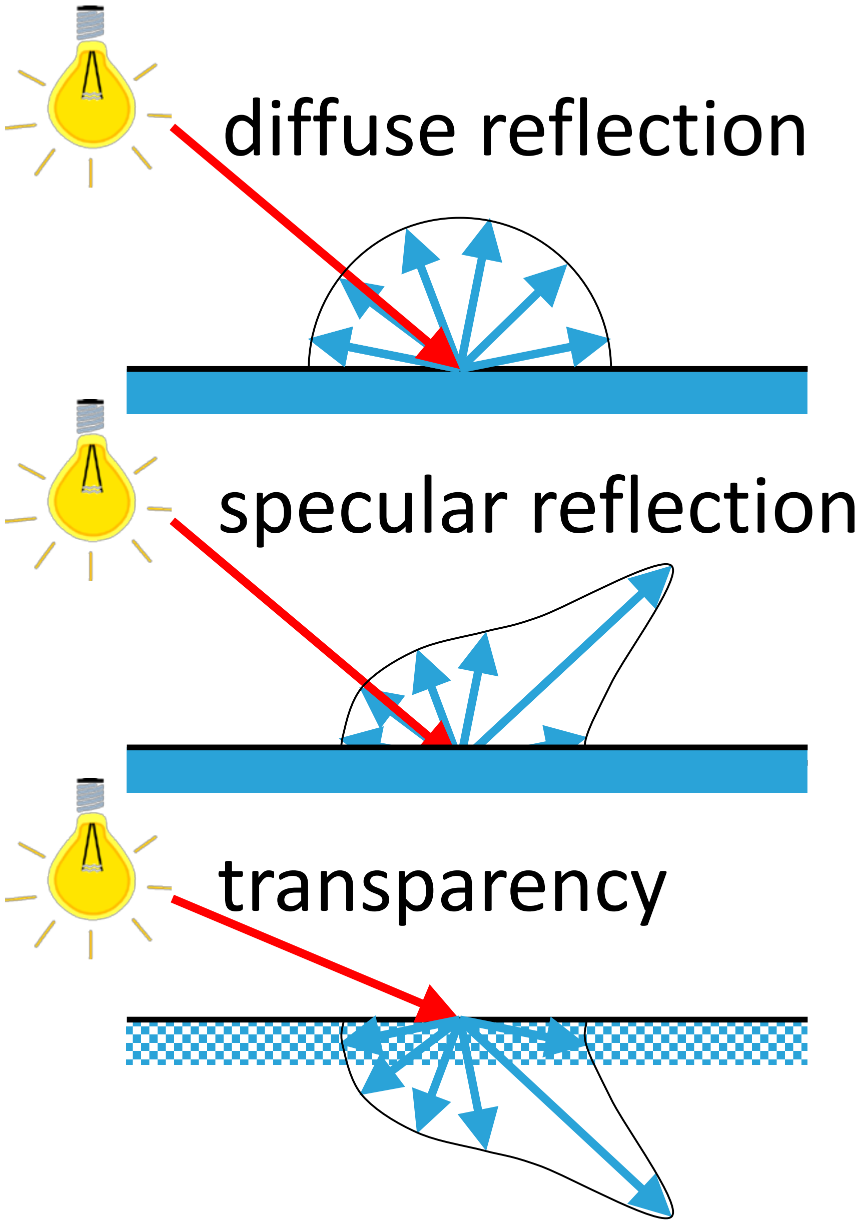
- Gouraud Shading
- Phong Shading

Light Sources

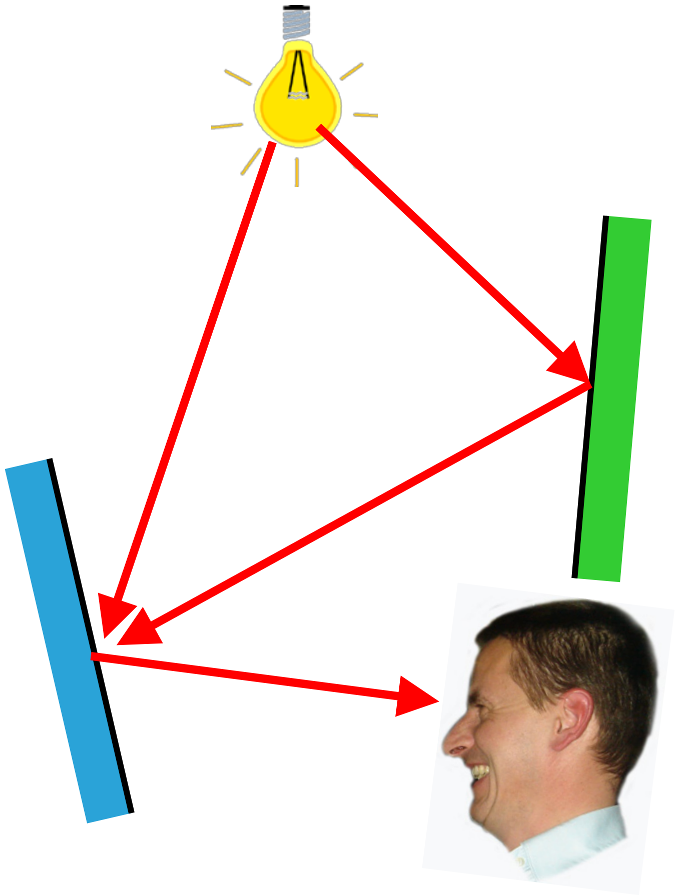
- directional light
- point light source (sometimes directional)
- distributed light source (“area light source”)



Surface Lighting Effects



reflections from other surfaces



Basic Illumination Models

models are empirical

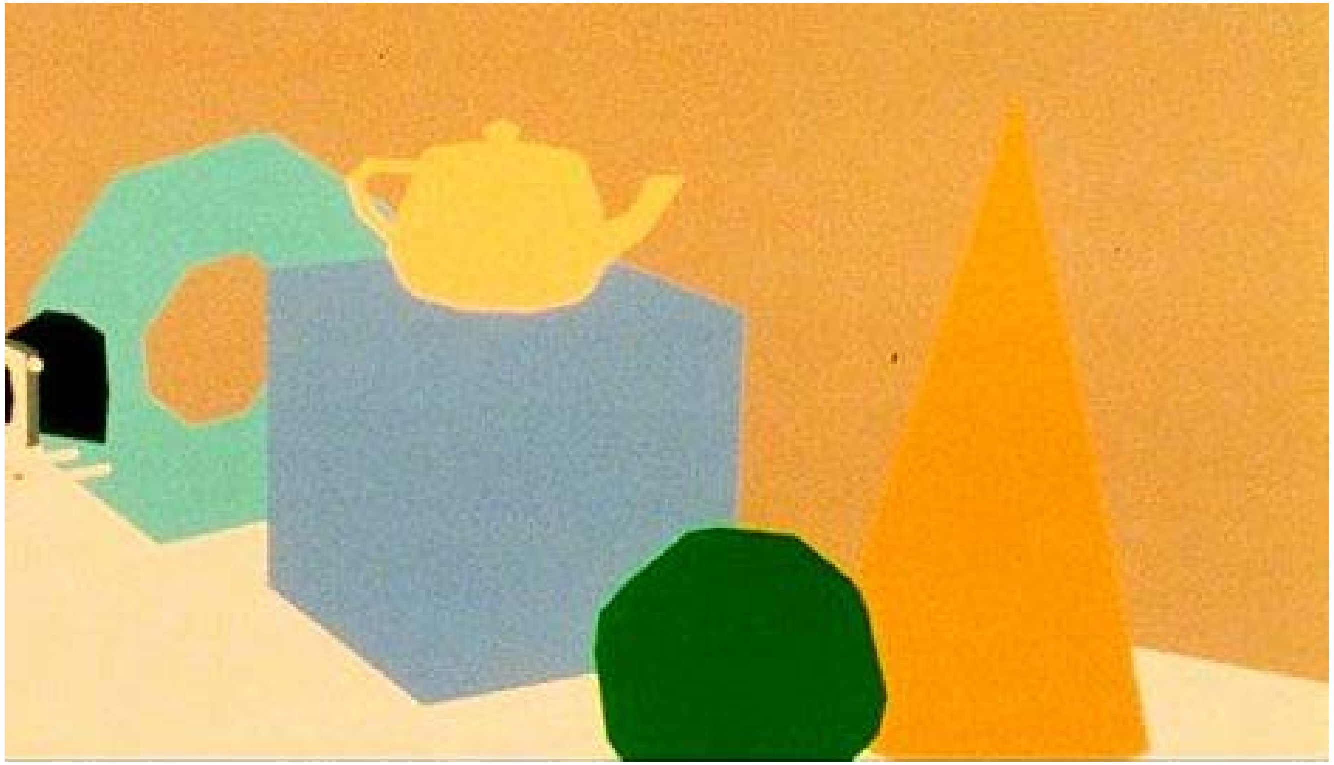
lighting calculations include

- surface properties (glossy, matte, opaque,...)
- background lighting conditions (ambient light)
- light-source specification
- reflection, absorption
- ...

Ambient Light Reflection

- ambient light (background light) $\mathbf{I_a}$
- constant over a surface
- independent of viewing direction
- diffuse-reflection coefficient $\mathbf{k_d}$ $(0 \leq \mathbf{k_d} \leq 1)$
- approximation of global diffuse lighting effects

$$L_{\text{ambdiff}} = k_{\mathbf{d}} I_{\mathbf{a}}$$



shaded surfaces generate a spatial impression

the flatter light falls on a surface,
the darker it will appear

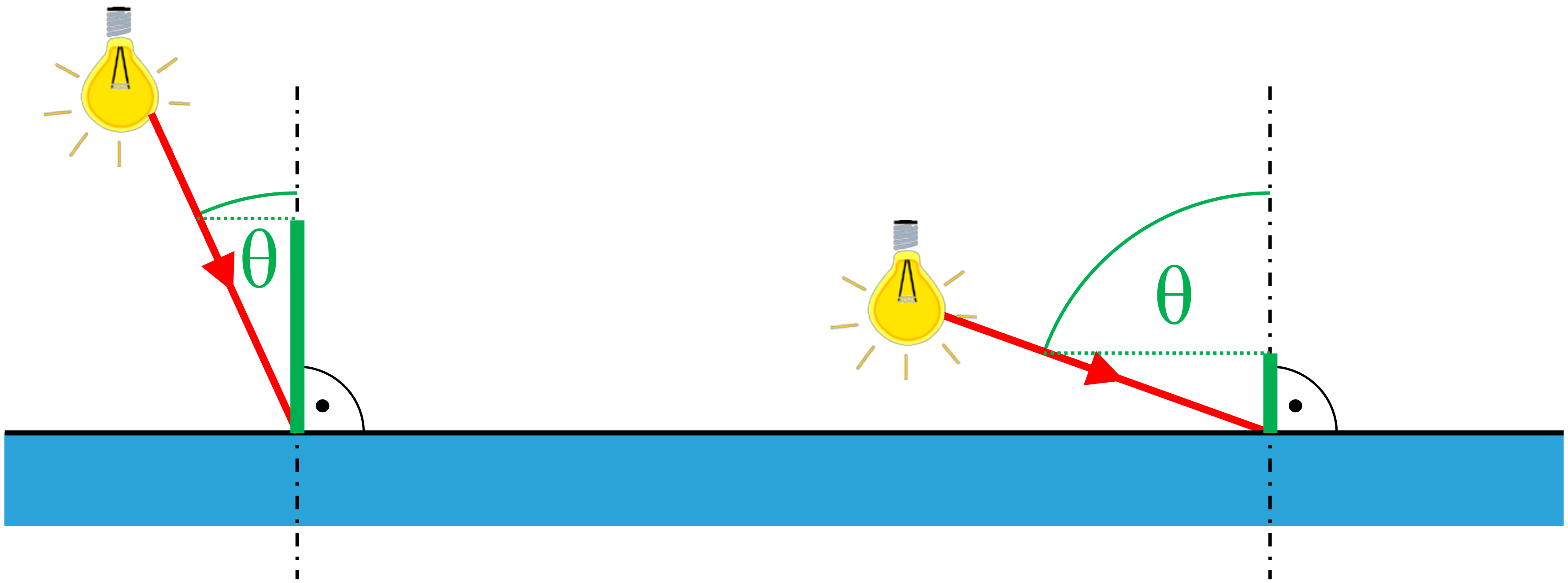
therefore:

we need the incident light direction

or

the position of the (point) light source

Lambert's Law



$L = I \cdot \cos\theta$

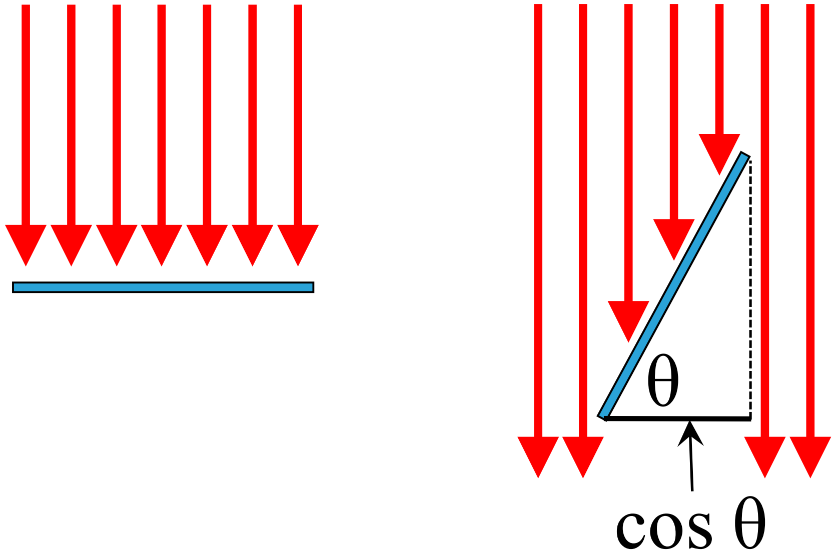
when considering the material:

$$\boxed{L = k_d \cdot I \cdot \cos\theta}$$

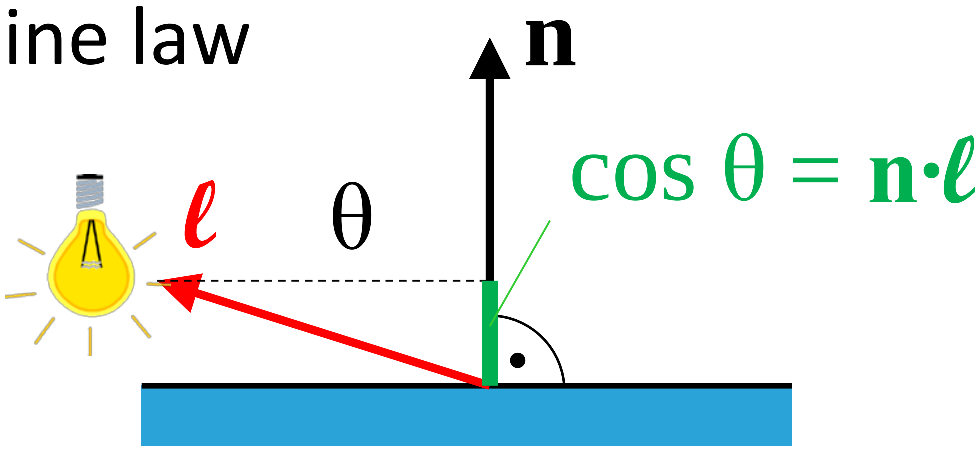
I … light source intensity
L … pixel color
k_d … diffuse coefficient

Lambertian (Diffuse) Reflection

- for ideal diffuse reflectors (Lambertian reflectors)
- brightness depends on orientation of the surface:



- Lambert's cosine law

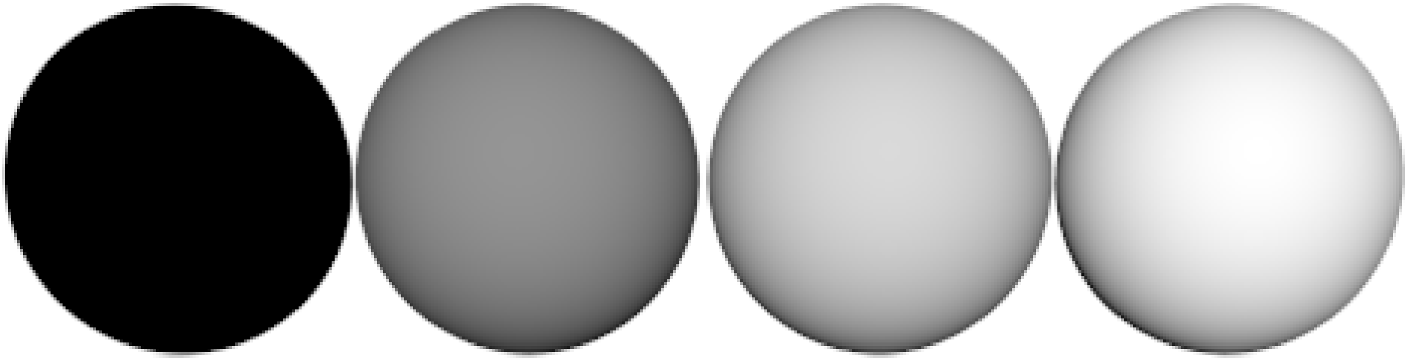


$$L_{\text{diff}} = k_d \cdot I \cdot (\mathbf{n} \cdot \boldsymbol{\ell})$$

Diffuse Reflection Coefficient

varying $k_{\mathbf{d}}$:

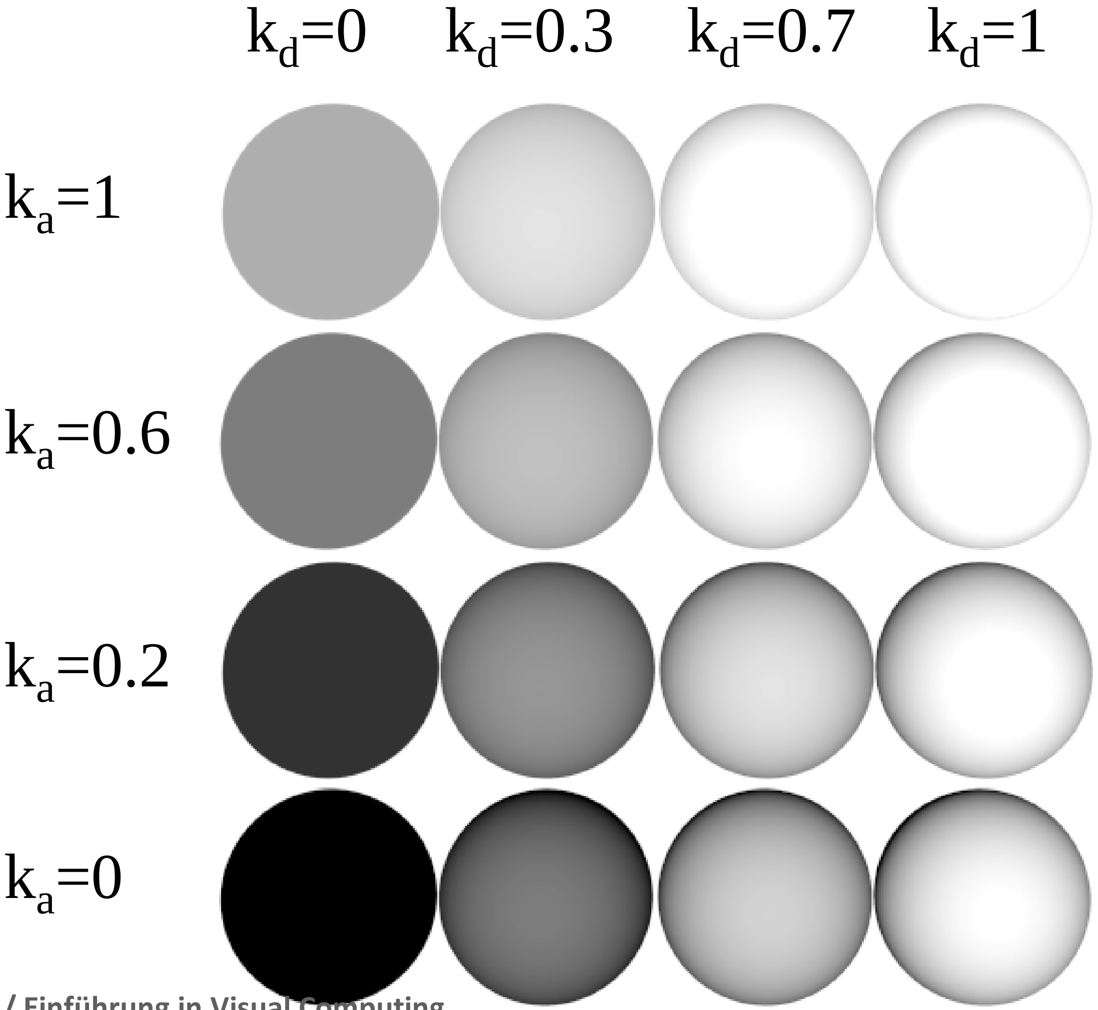
result for varying values of $k_{\mathbf{d}}$ ($I_{\mathbf{a}} = 0$)



k_d=0 k_d=0.3 k_d=0.7 k_d=1

Ambient + Diffuse Reflection

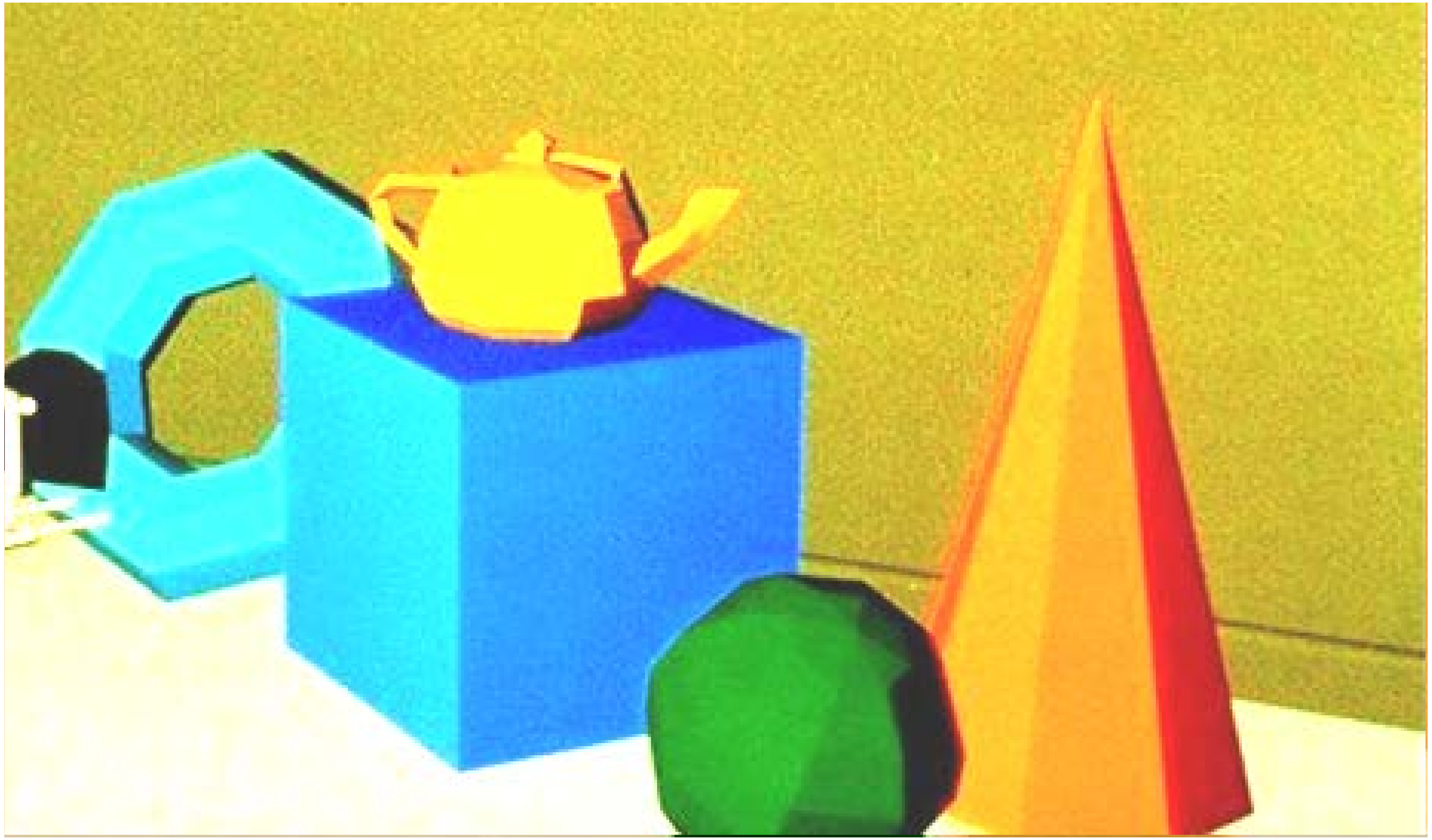
total diffuse reflection: $$L_{diff} = k_a I_a + k_d I(\mathbf{n} \cdot \ell)$$



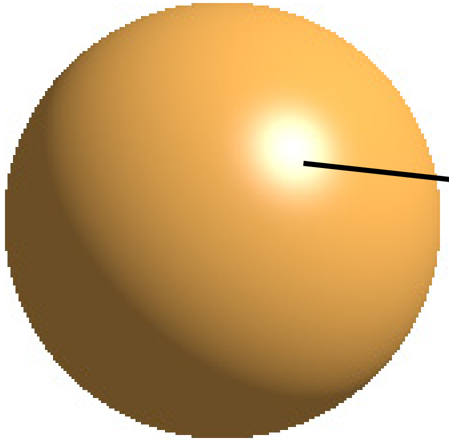
(sometimes extra $\mathrm{k}_{\mathbf{a}}$ for ambient light)

Ambient plus Diffuse Reflection

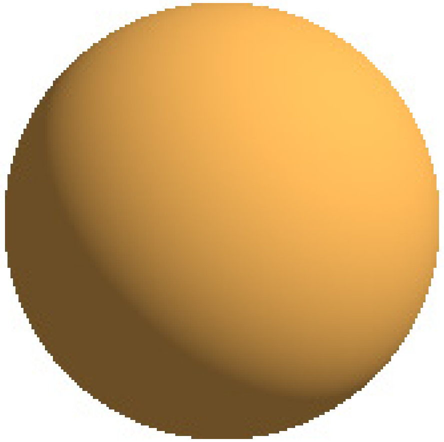
total diffuse reflection: $$L_{diff} = k_a I_a + k_d I(\mathbf{n} \cdot \boldsymbol{\ell})$$



Specular Highlights

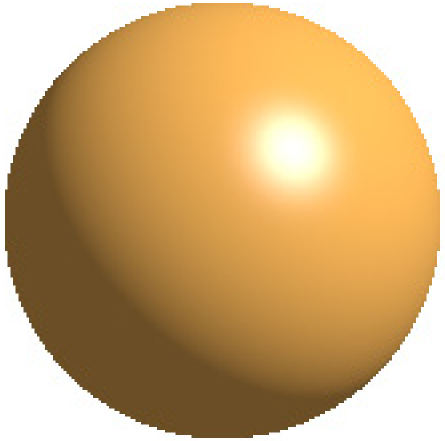


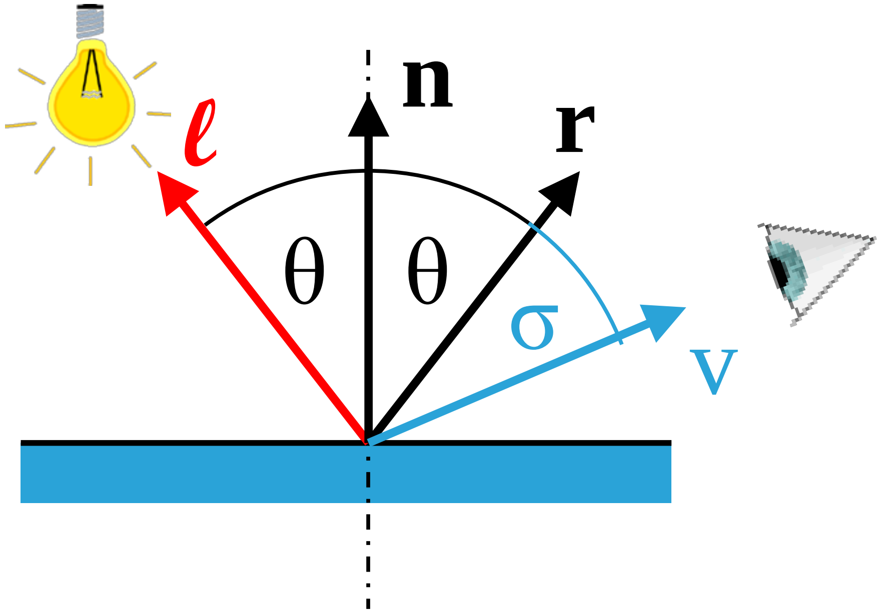
this area must be lighter than the shading model calculates, because the light source is reflected directly into the viewer's eye



Specular Reflection Model

reflection of incident light around specular-reflection angle:



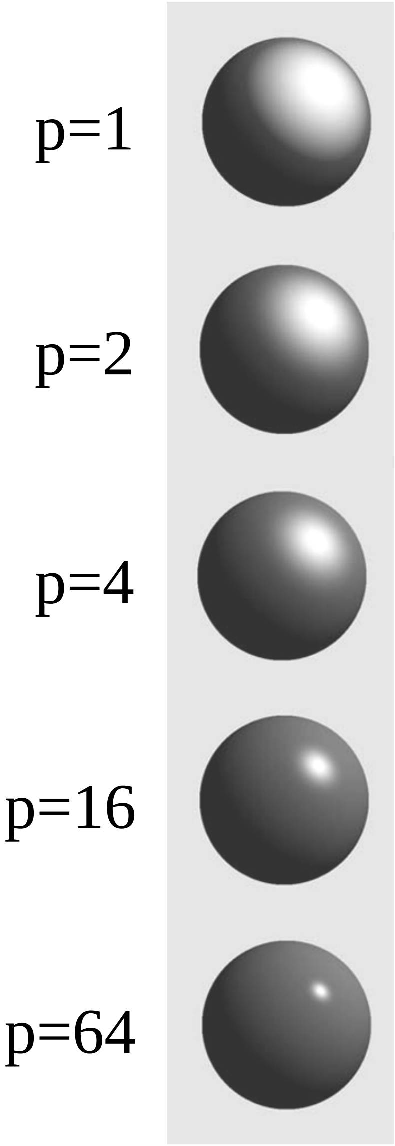


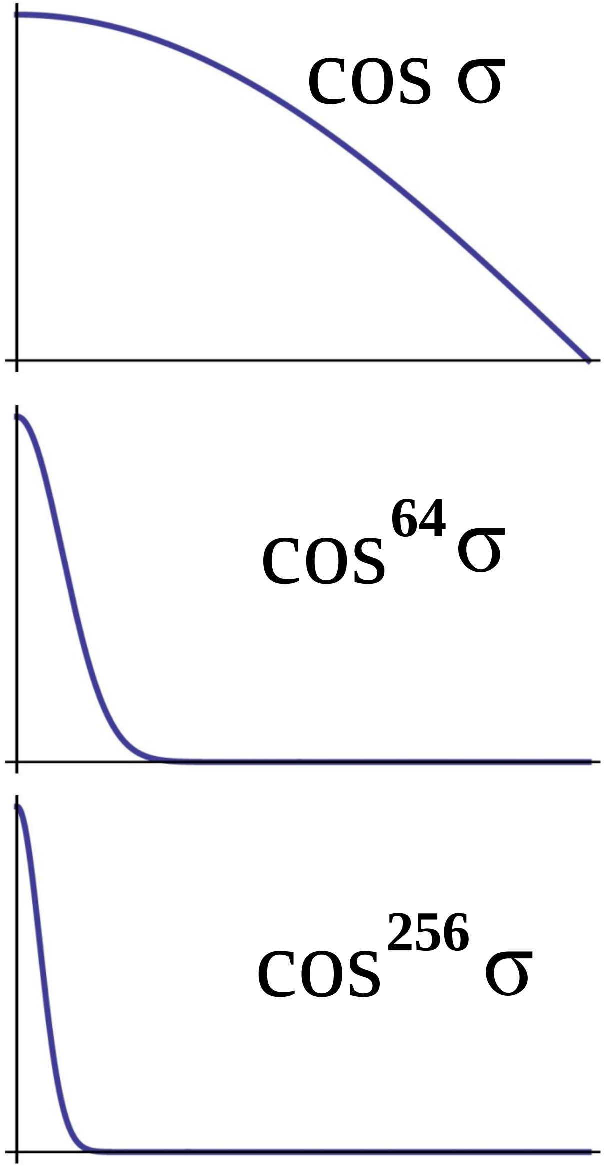
$\Rightarrow$ empirical Phong model

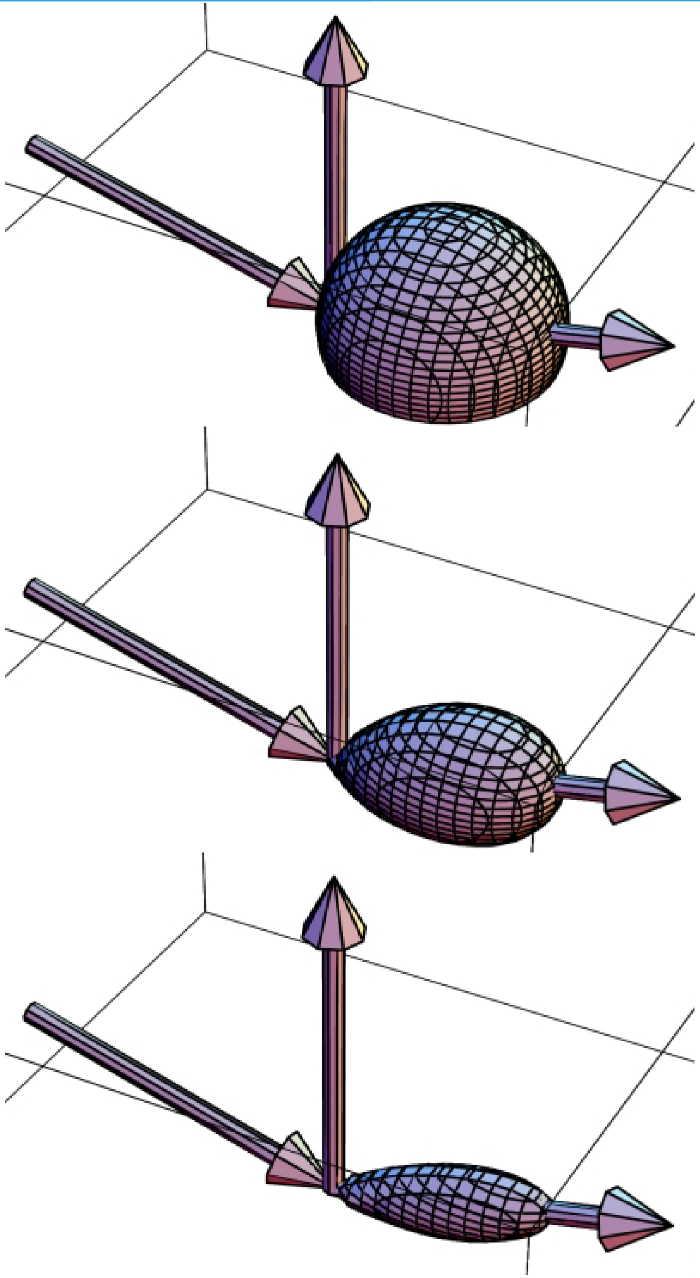
$$L_{spec} = k_s \cdot I \cdot \cos^p \sigma$$

Specular Reflection Coefficient p

$\cos^p \sigma$



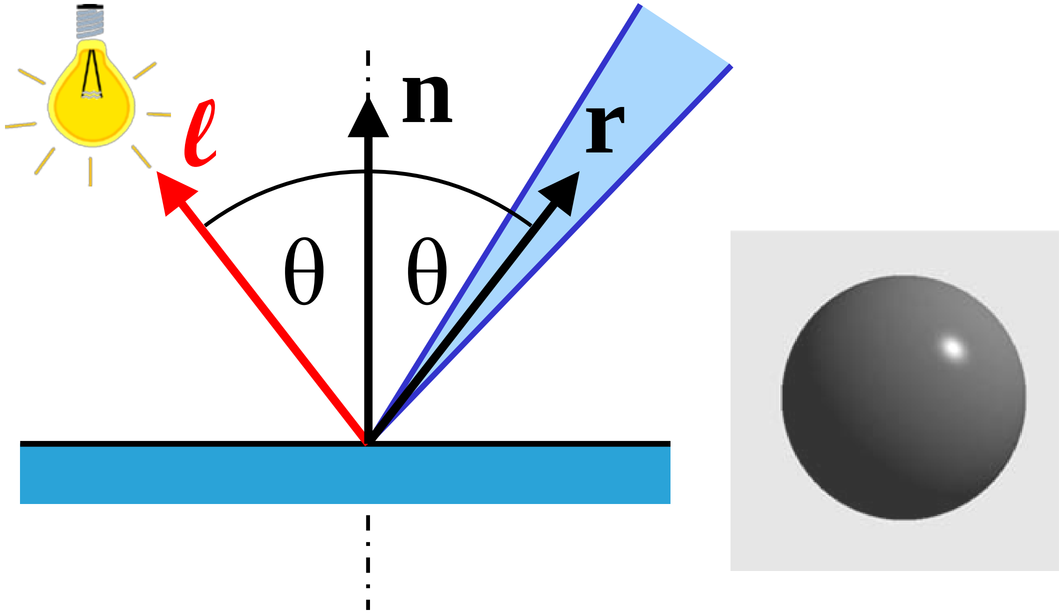




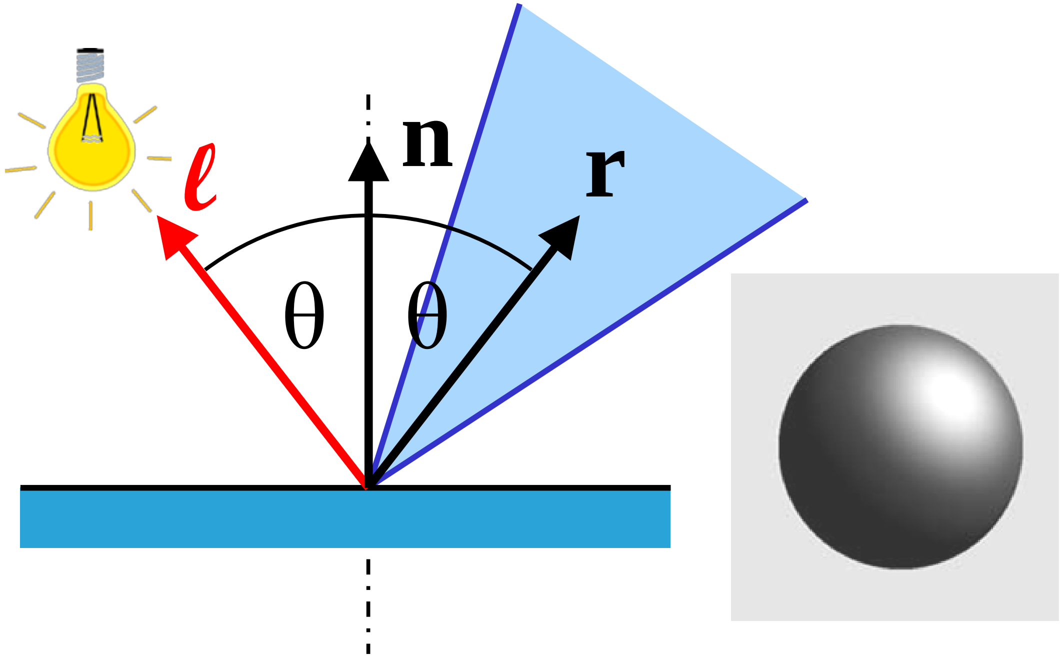
Specular Reflection Coefficient

empirical Phong model $$L_{spec} = k_s \cdot I \cdot \cos^p \sigma$$

p large ⇒ shiny surface

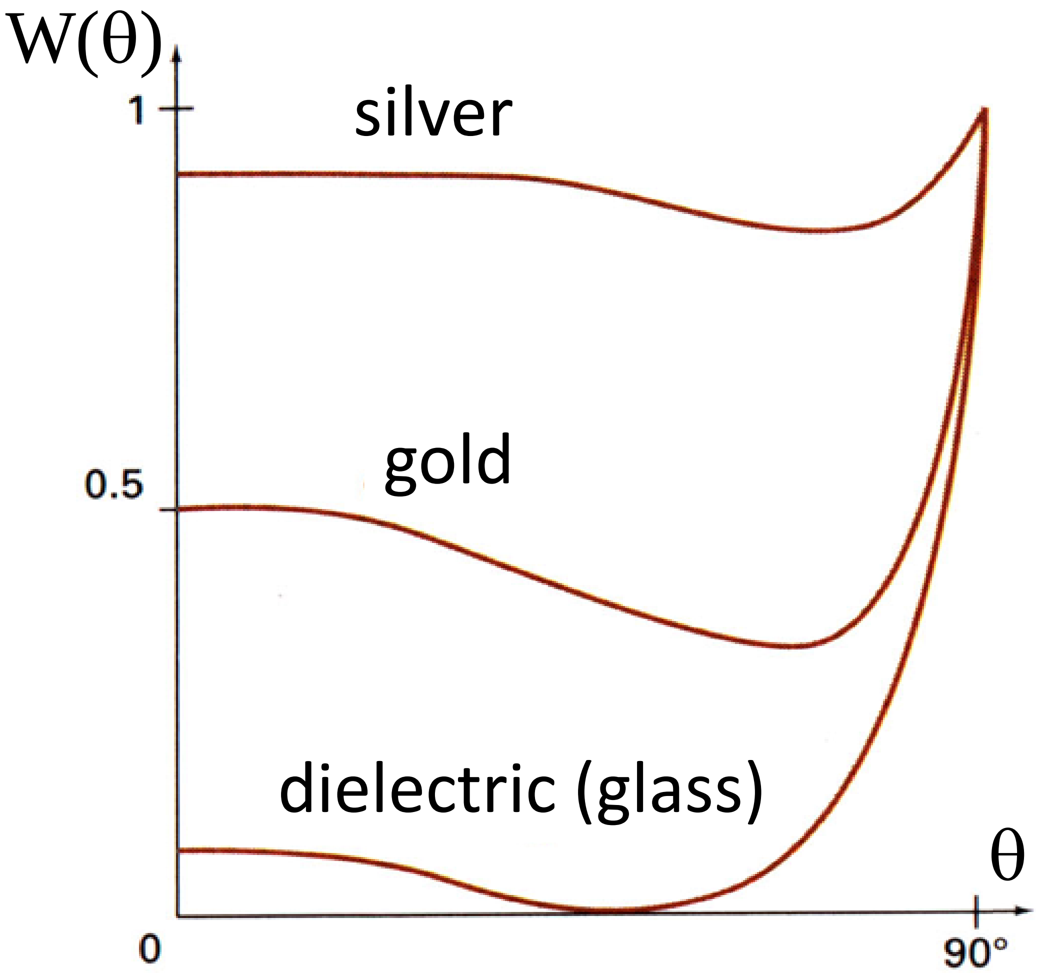


p small ⇒ dull surface



Fresnel Specular Reflection Coefficient

Fresnel's laws of reflection use specular reflection coefficient $W(\theta)$

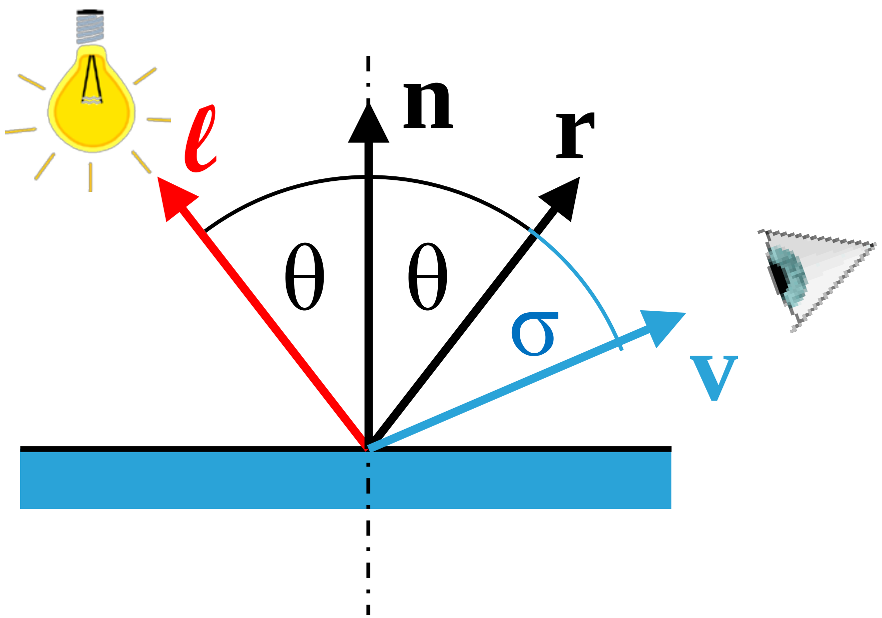


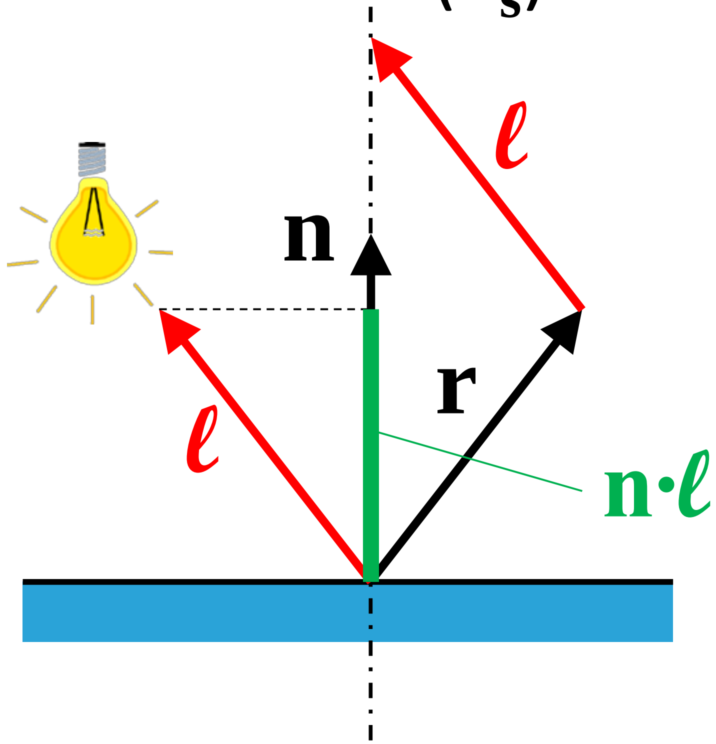
$$L_{spec} = W(\theta)\, I_{\ell} \cos^{p}\sigma$$

specular reflection coefficient as a function of angle of incidence for different materials

Simple Specular Reflection

$W(\theta) \approx$ constant for many opaque materials (k_s)





$$L_{spec} = k_s \cdot I \cdot (\mathbf{v} \cdot \mathbf{r})^p$$

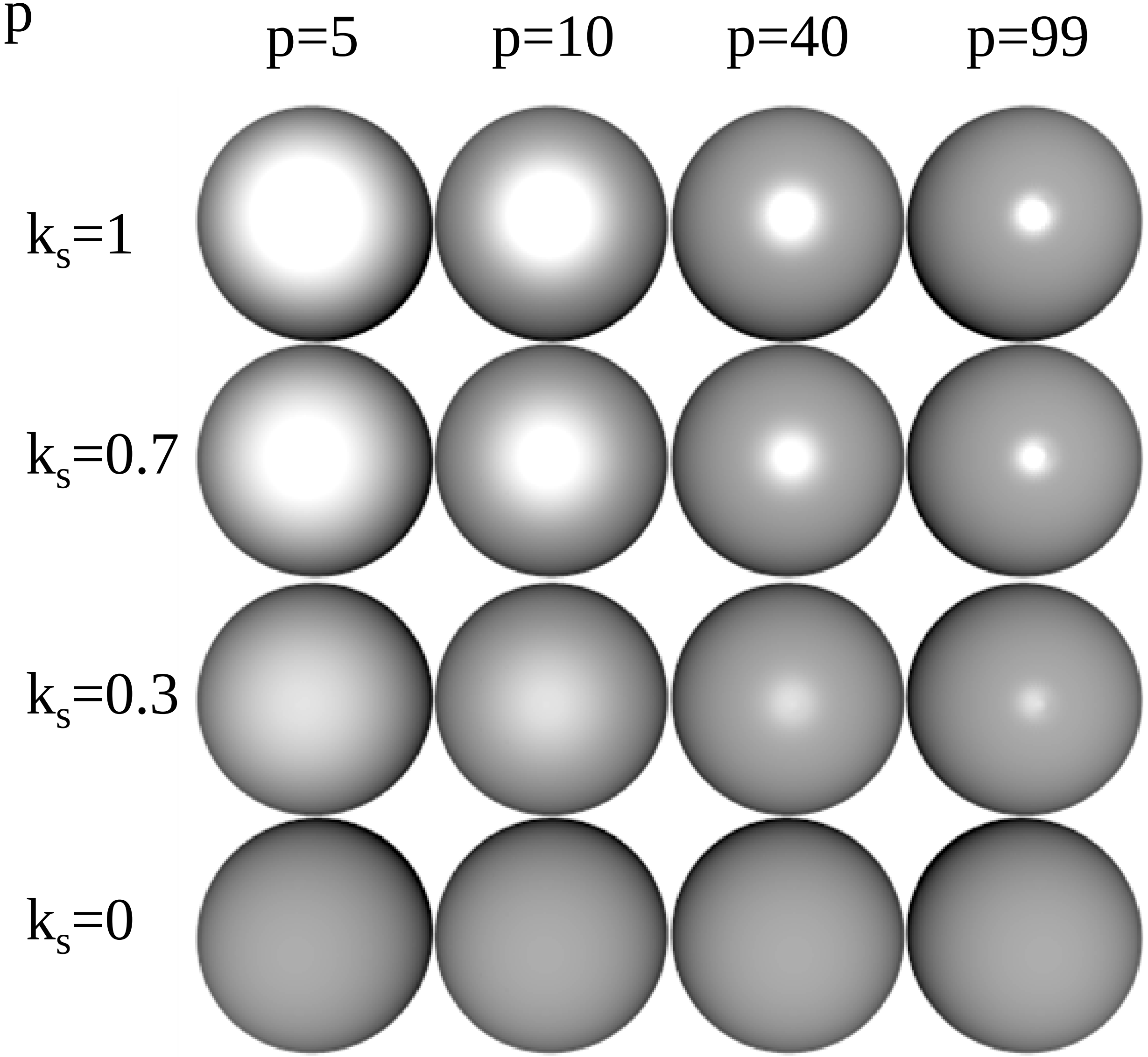
calculation of **r**:

$$\mathbf{r} + \boldsymbol{\ell} = (2\mathbf{n} \cdot \boldsymbol{\ell})\mathbf{n}$$

$$\mathbf{r} = (2\mathbf{n} \cdot \boldsymbol{\ell})\mathbf{n} - \boldsymbol{\ell}$$

Specular Reflection Results

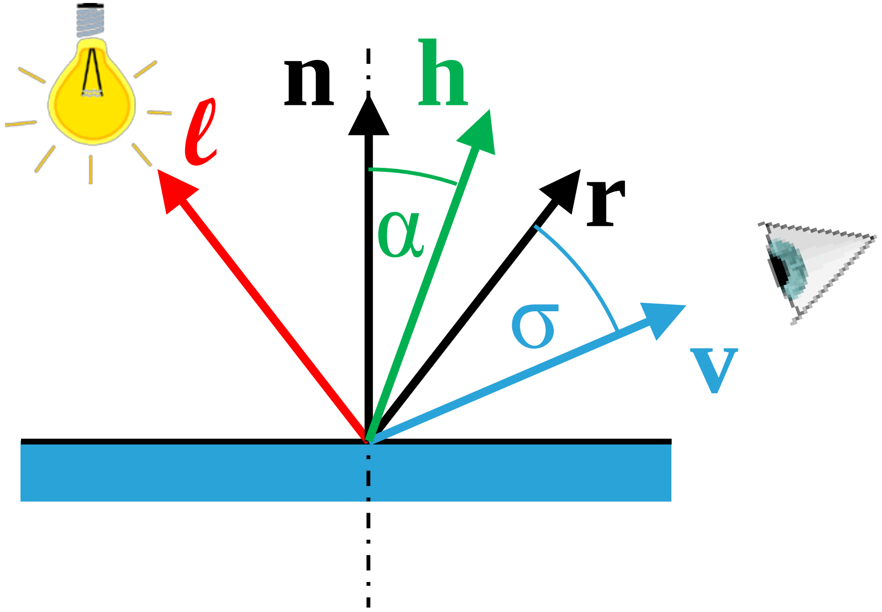
$$L_{spec} = k_s \cdot I \cdot (\mathbf{v} \cdot \mathbf{r})^p$$



Simplified Specular Reflection

simplified Phong model with halfway vector **h**

$$L_{spec} = k_s \cdot I \cdot (\mathbf{v} \cdot \mathbf{r})^p \quad \rightarrow \quad L_{spec} = k_s \cdot I \cdot (\mathbf{n} \cdot \mathbf{h})^p$$



$$\mathbf{h} = \frac{\boldsymbol{\ell} + \mathbf{v}}{\| \boldsymbol{\ell} + \mathbf{v} \|}$$

this revised model is called “Blinn-Phong Shading”

Diffuse and Specular Reflection

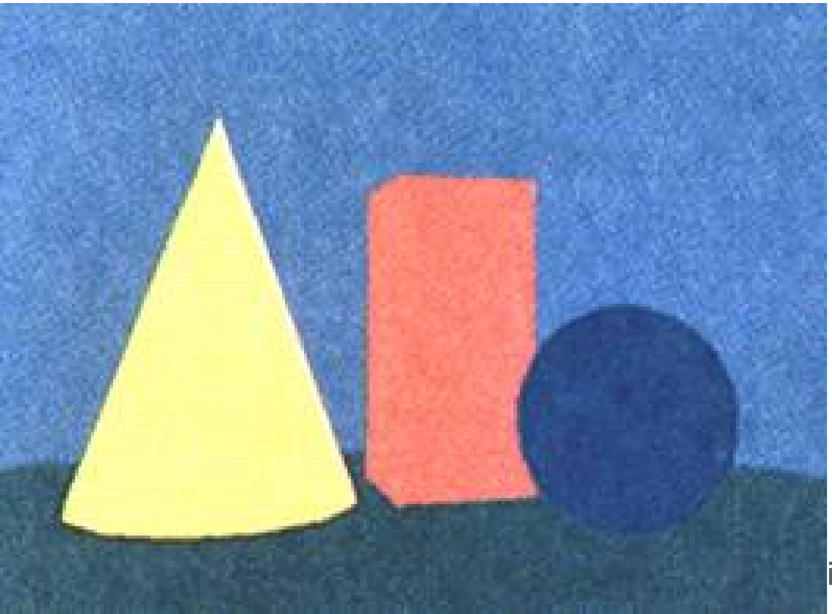
$$L = k_a I_a + \sum_{i=1}^{N} I_i \left[k_d (\mathbf{n} \cdot \boldsymbol{\ell}_i) + k_s (\mathbf{n} \cdot \mathbf{h}_i)^p \right]$$

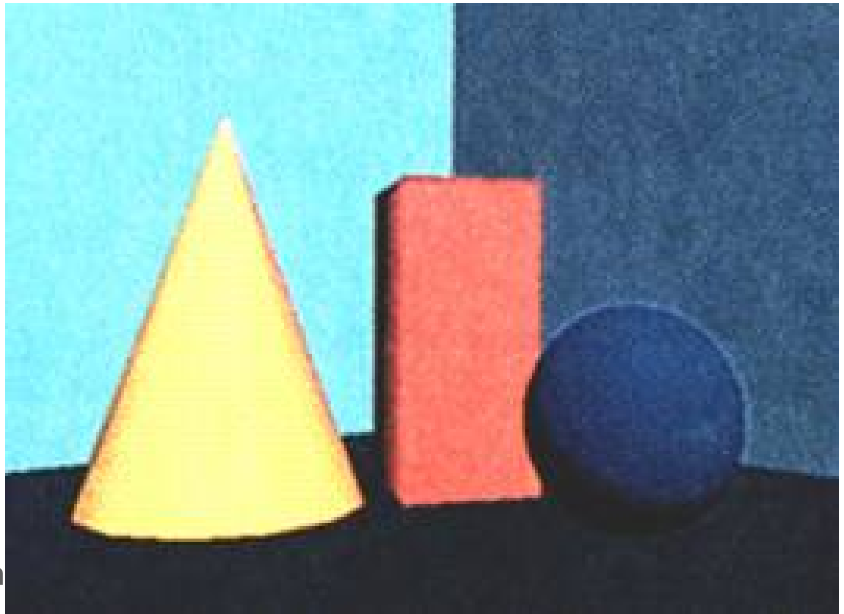
© James Tasker © Tong Lai Yu

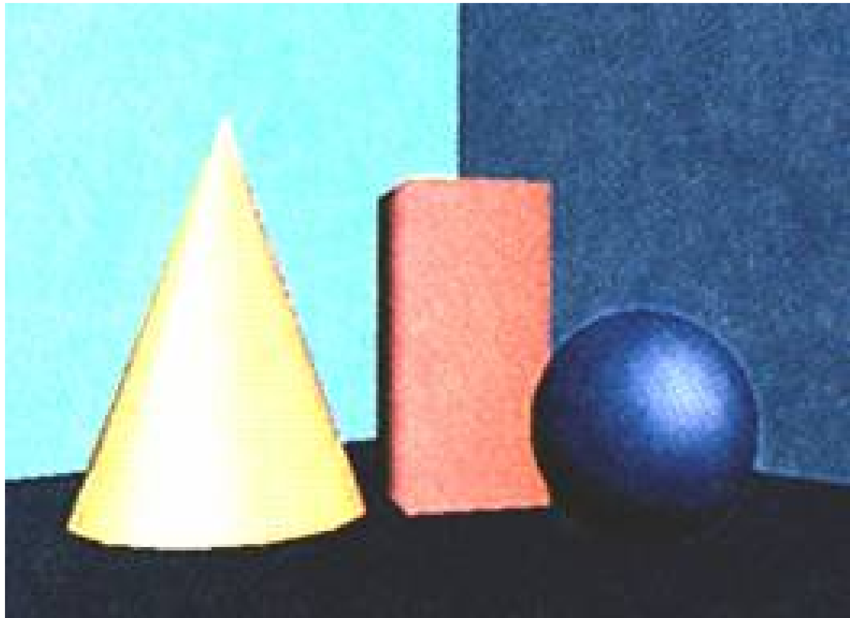
ambient

ambient + diffuse

ambient + diffuse + specular



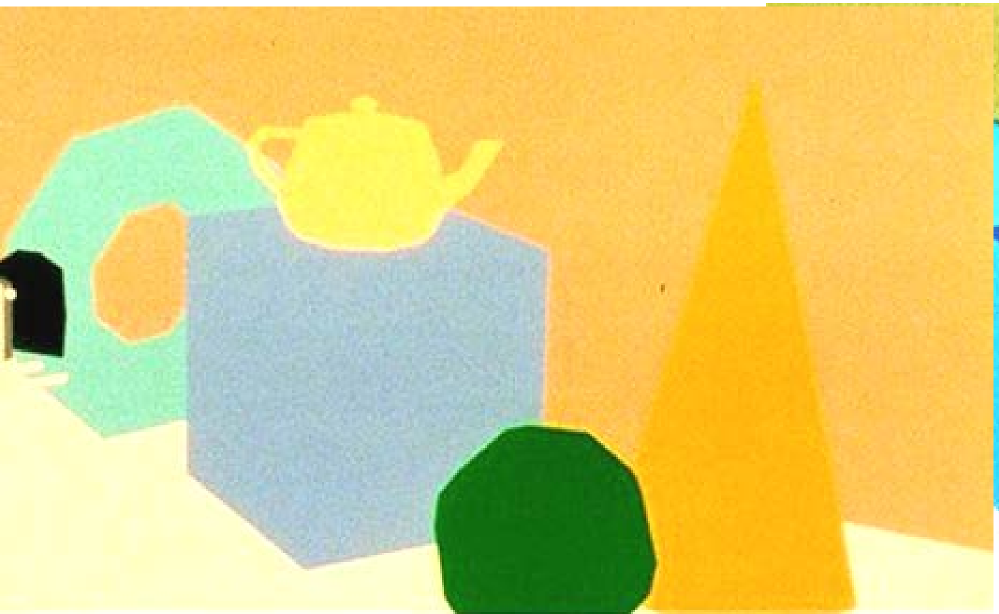




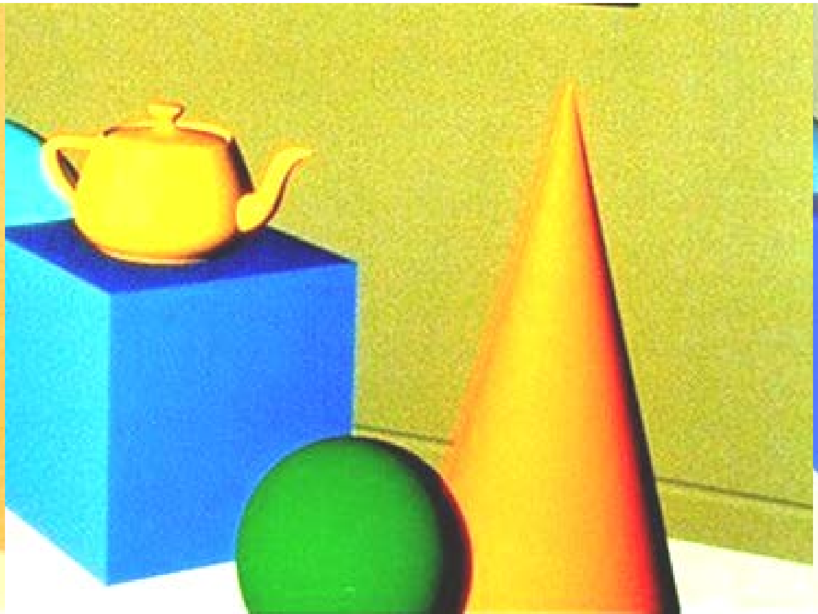
Diffuse and Specular Reflection

$$L = k_a I_a + \sum_{i=1}^{N} I_i \left[k_d (\mathbf{n} \cdot \boldsymbol{\ell}_\mathbf{i}) + k_s (\mathbf{n} \cdot \mathbf{h}_\mathbf{i})^p \right]$$

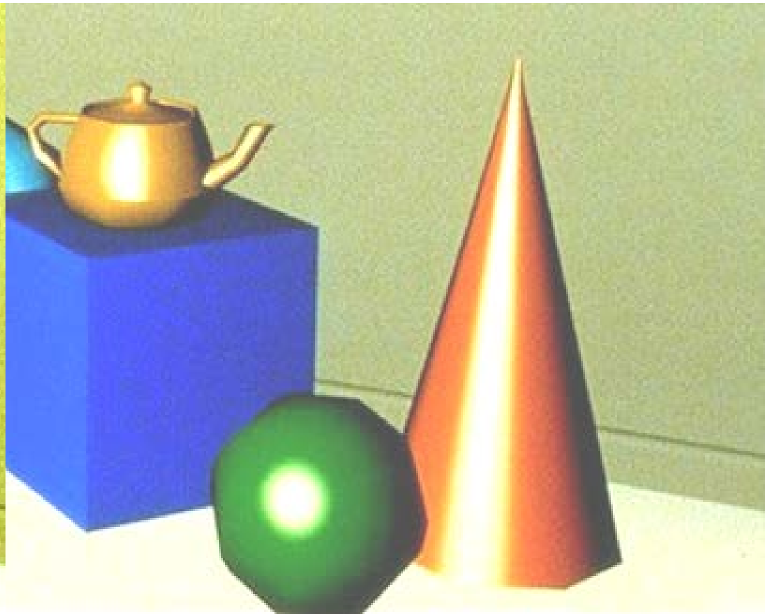
ambient



ambient + diffuse



ambient + diffuse
+ specular



Other Aspects

- intensity attenuation with distance
- anisotropic light sources
- transparency (Snell's law)
- atmospheric effects
- shadows
- …

Polygon-Rendering Methods

application of illumination model to polygon rendering

constant-intensity shading (flat shading)

= single intensity for each polygon

flat

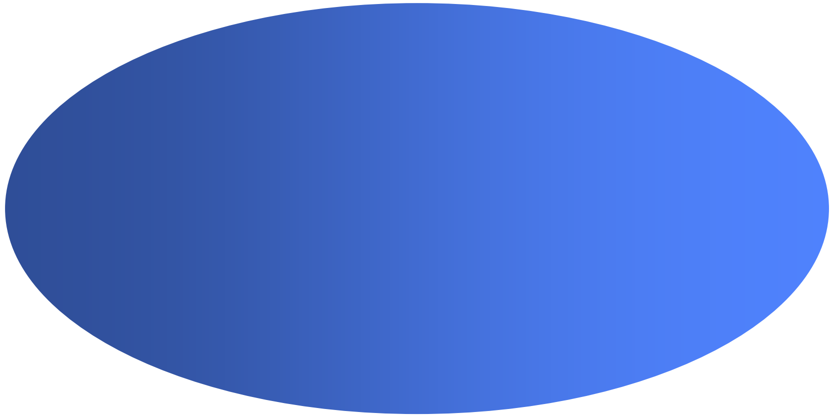
Gouraud

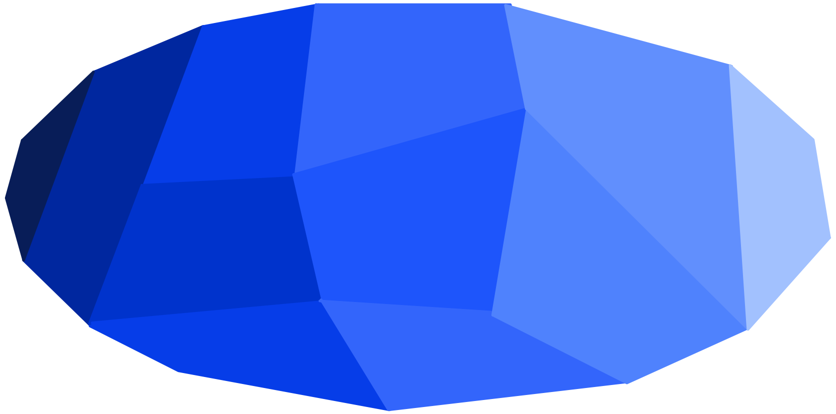
Polygon Shading: Interpolation

the shading of a polygon is not constant, because it normally is only an approximation of the real surface ⇒ **interpolation**

Gouraud shading: intensities

Phong shading: normal vectors

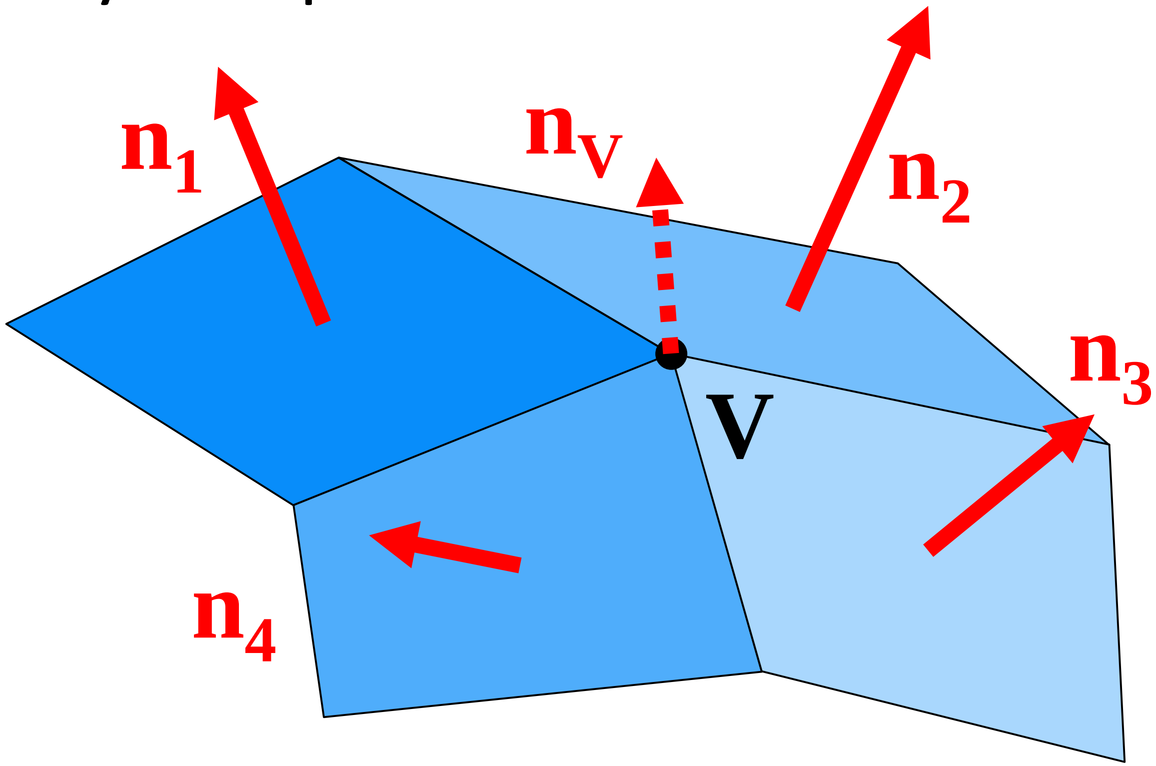




Gouraud Shading Overview

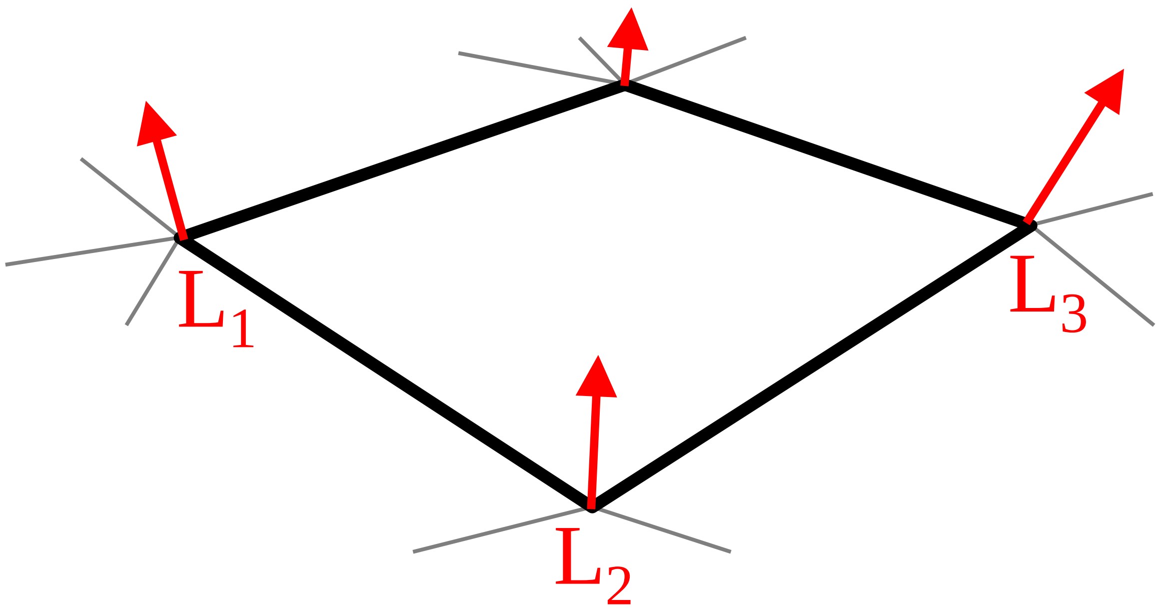
intensity-interpolation:

a. determine average unit normal vector at each polygon vertex
b. apply illumination model to each vertex
c. linearly interpolate vertex intensities



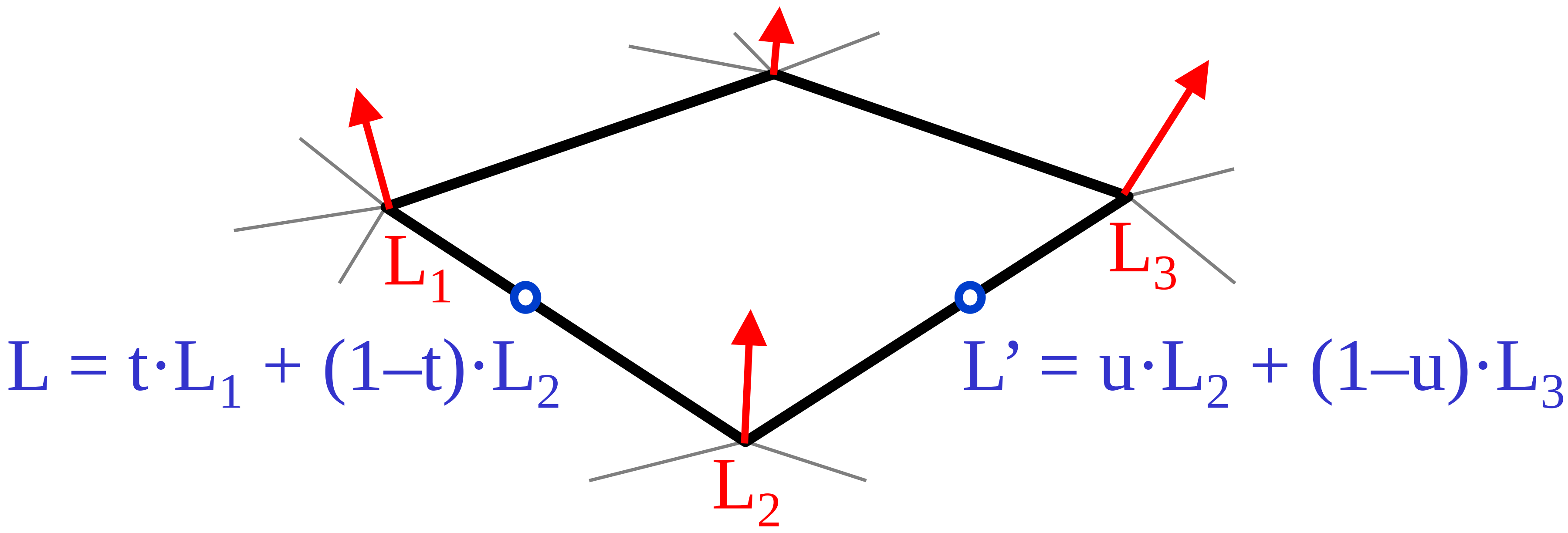
$$\mathbf{n}_V = \frac{\sum_{k=1}^{N} \mathbf{n}_k}{\left\| \sum_{k=1}^{N} \mathbf{n}_k \right\|}$$

Gouraud Shading



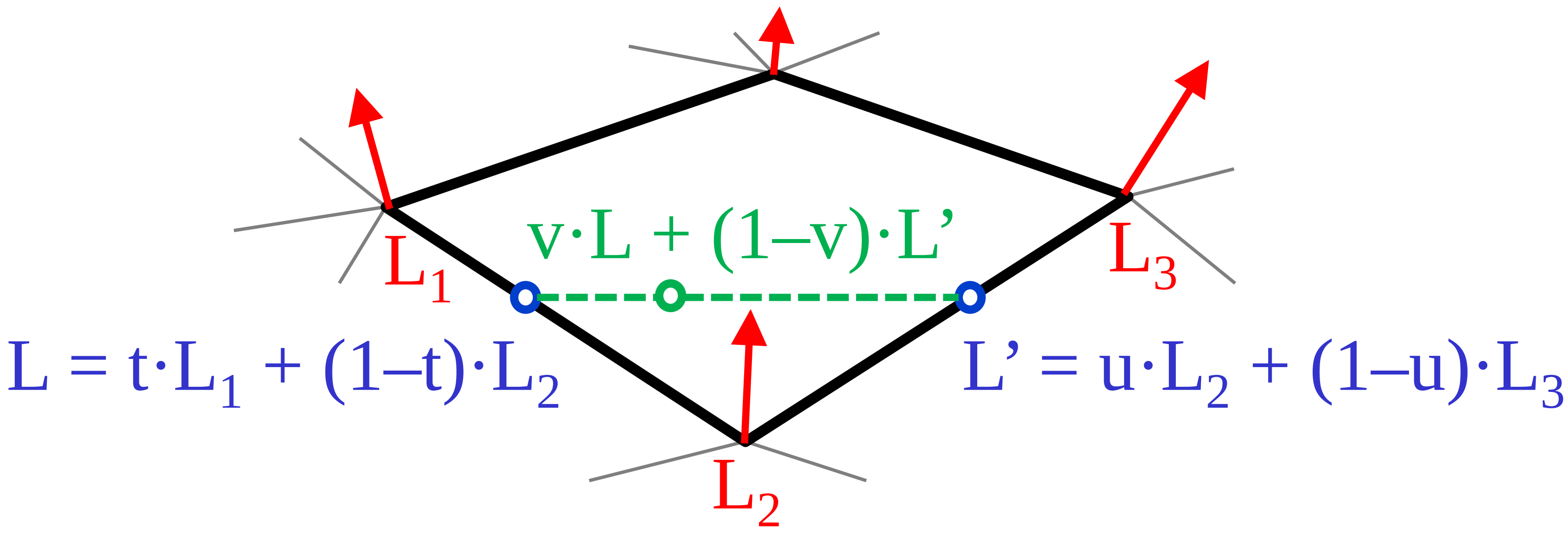
1. find normal vectors at corners and calculate shading (intensities) there: L_i

Gouraud Shading



1. find normal vectors at corners and calculate shading (intensities) there: L_i
2. interpolate intensities along edges linearly: L, L'

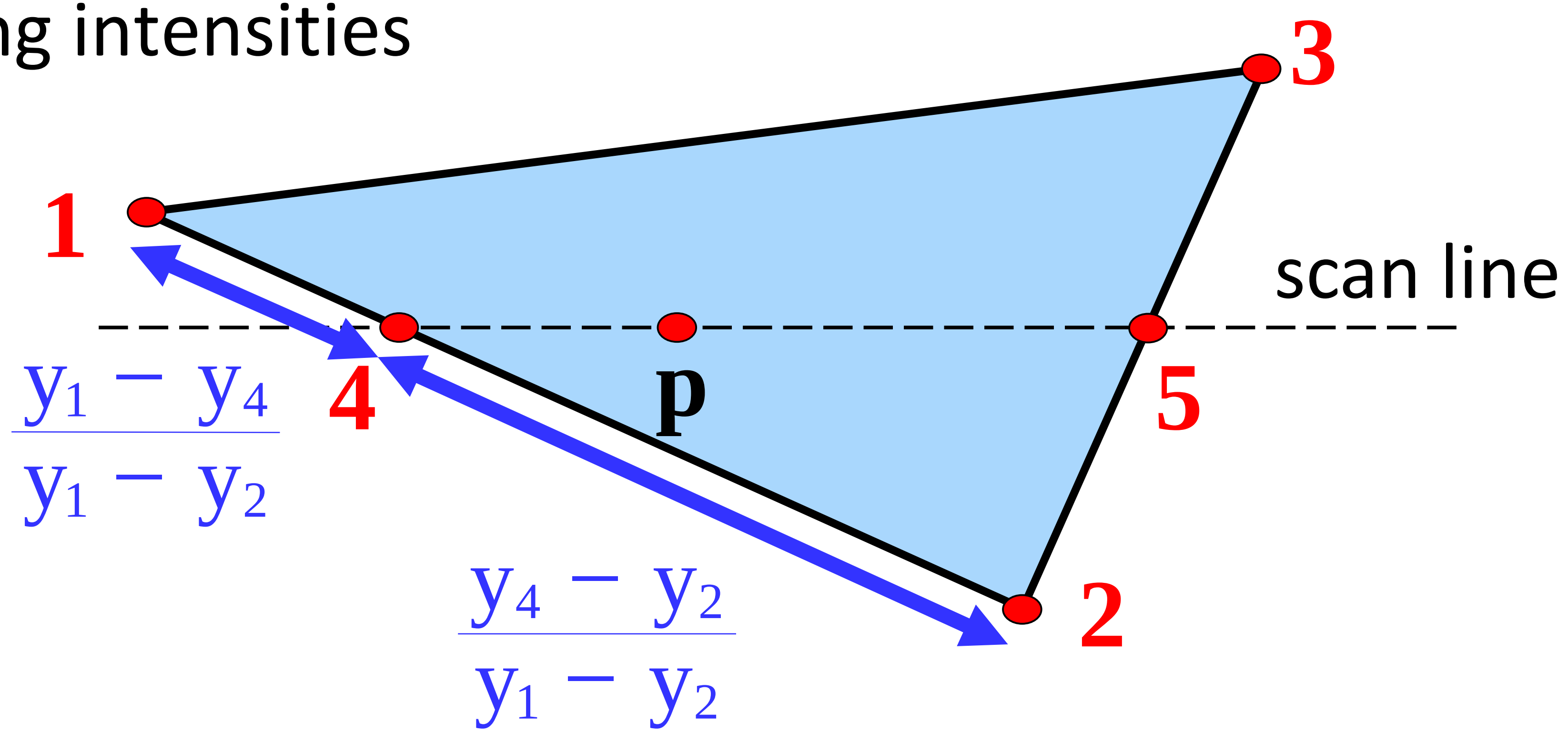
Gouraud Shading



1. find normal vectors at corners and calculate shading (intensities) there: L_i
2. interpolate intensities along edges linearly: L, L'
3. interpolate intensities along scanlines linearly: L_p

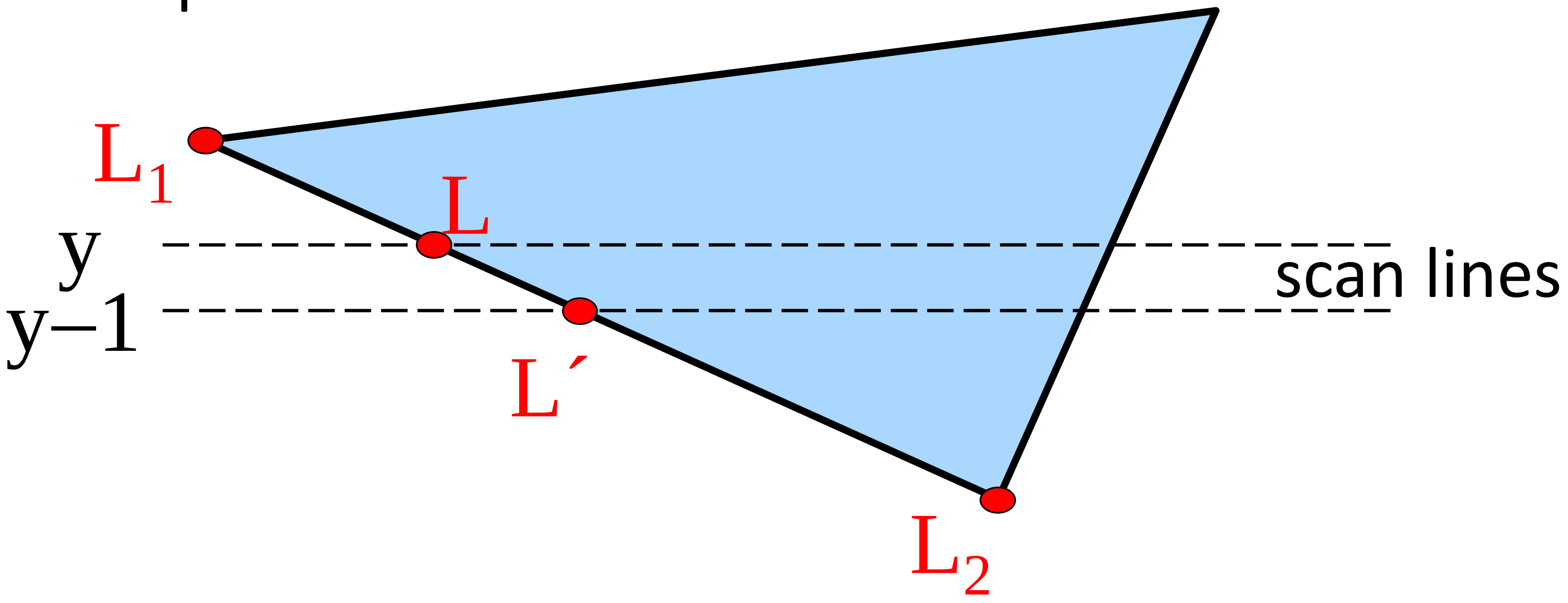
Gouraud Shading

interpolating intensities



$$L_4 = \frac{y_4 - y_2}{y_1 - y_2} L_1 + \frac{y_1 - y_4}{y_1 - y_2} L_2 \qquad L_p = \frac{x_5 - x_p}{x_5 - x_4} L_4 + \frac{x_p - x_4}{x_5 - x_4} L_5$$

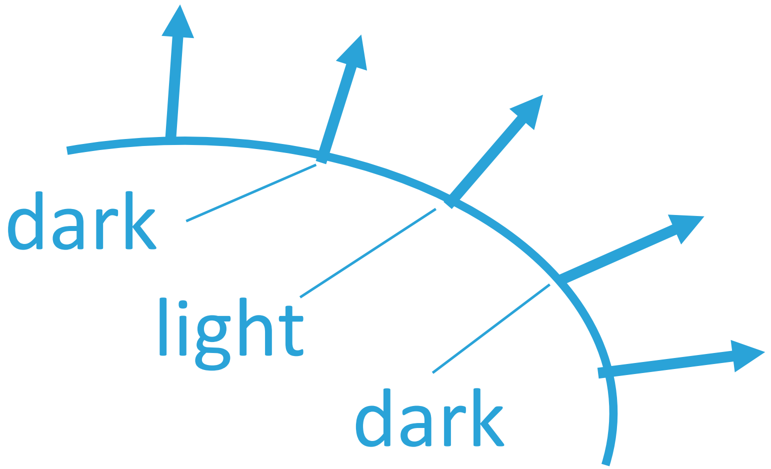
incremental update

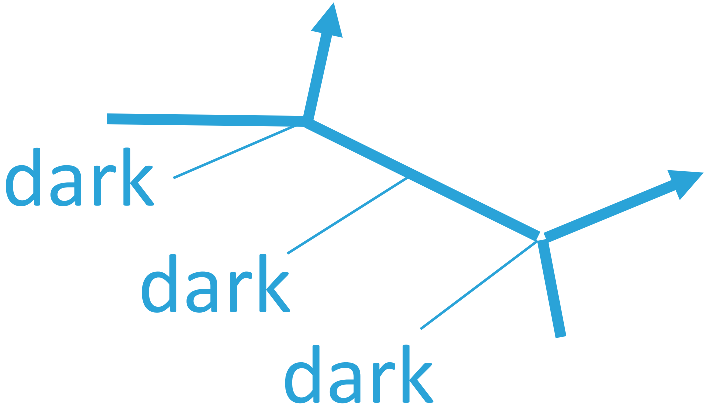


$$L = \frac{y - y_2}{y_1 - y_2} L_1 + \frac{y_1 - y}{y_1 - y_2} L_2 \qquad L' = L + \frac{L_2 - L_1}{y_1 - y_2}$$

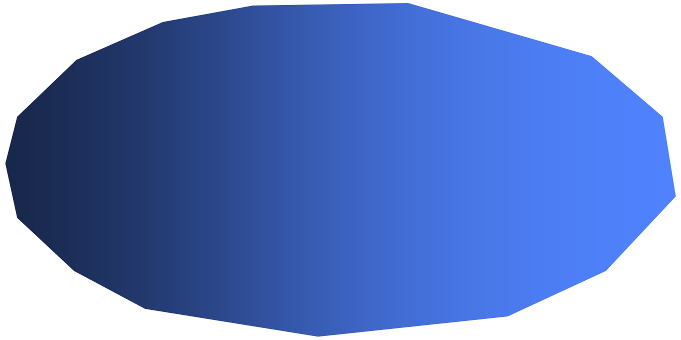
Problems of Gouraud Shading

highlights can get lost or grow





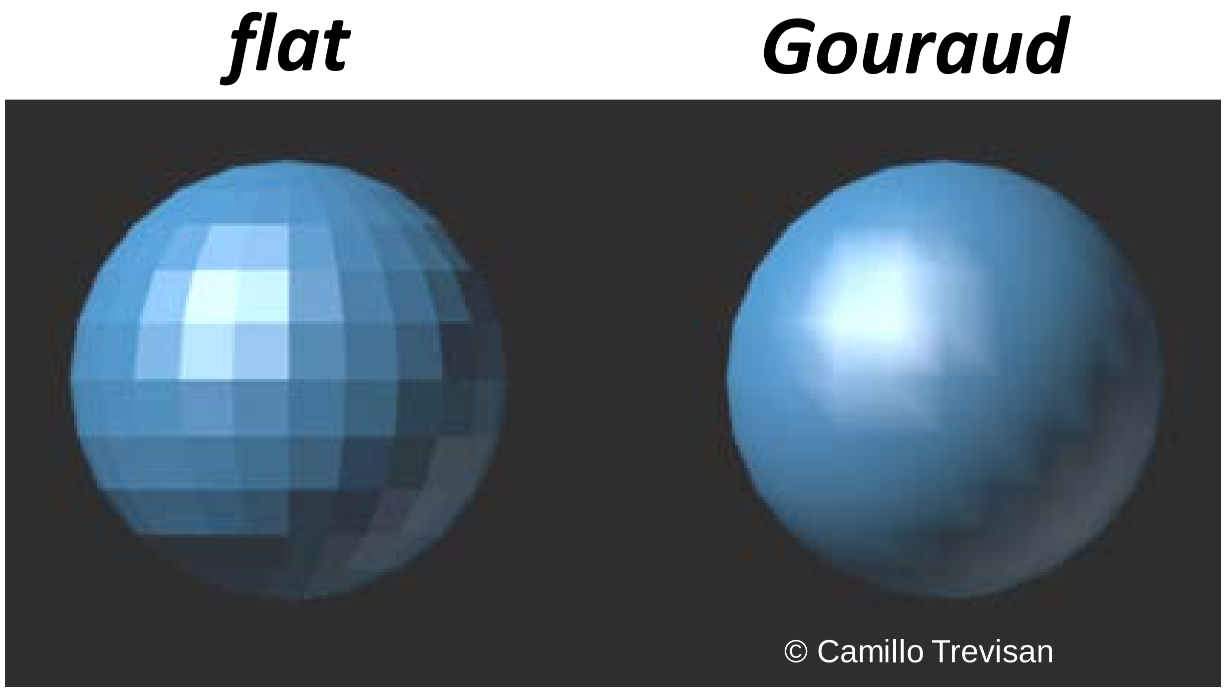
corners on silhouette remain



Mach band effect is visible at some edges

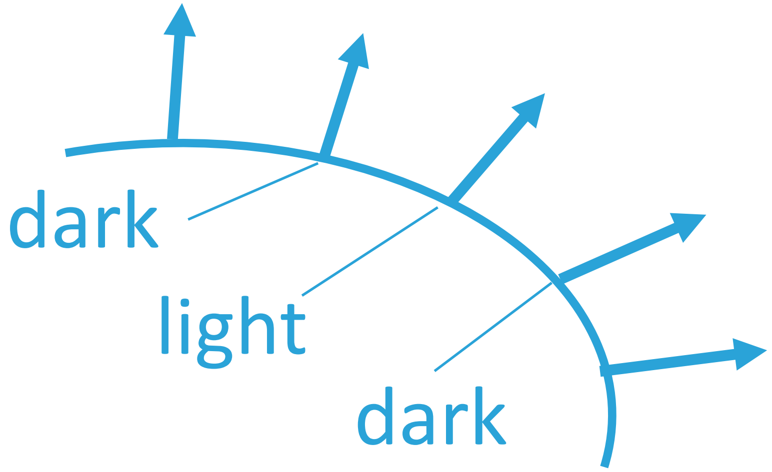
Gouraud Shading Results

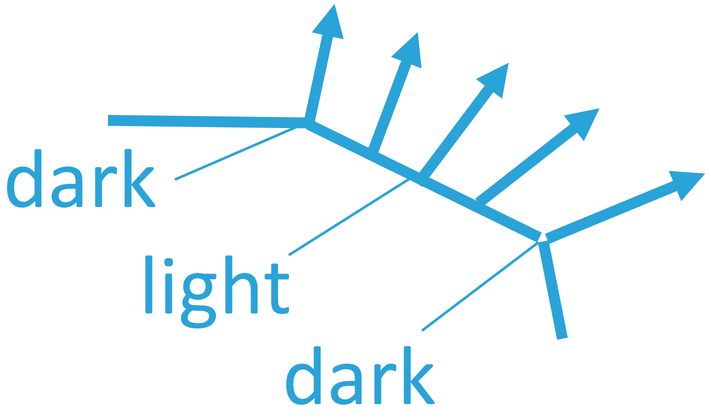
- no intensity discontinuities
- Mach bands due to linear intensity interpolation
- problems with highlights



Phong Shading

instead of intensities the normal vectors are interpolated, and for every point the shading calculation is performed separately





Phong Shading Principle

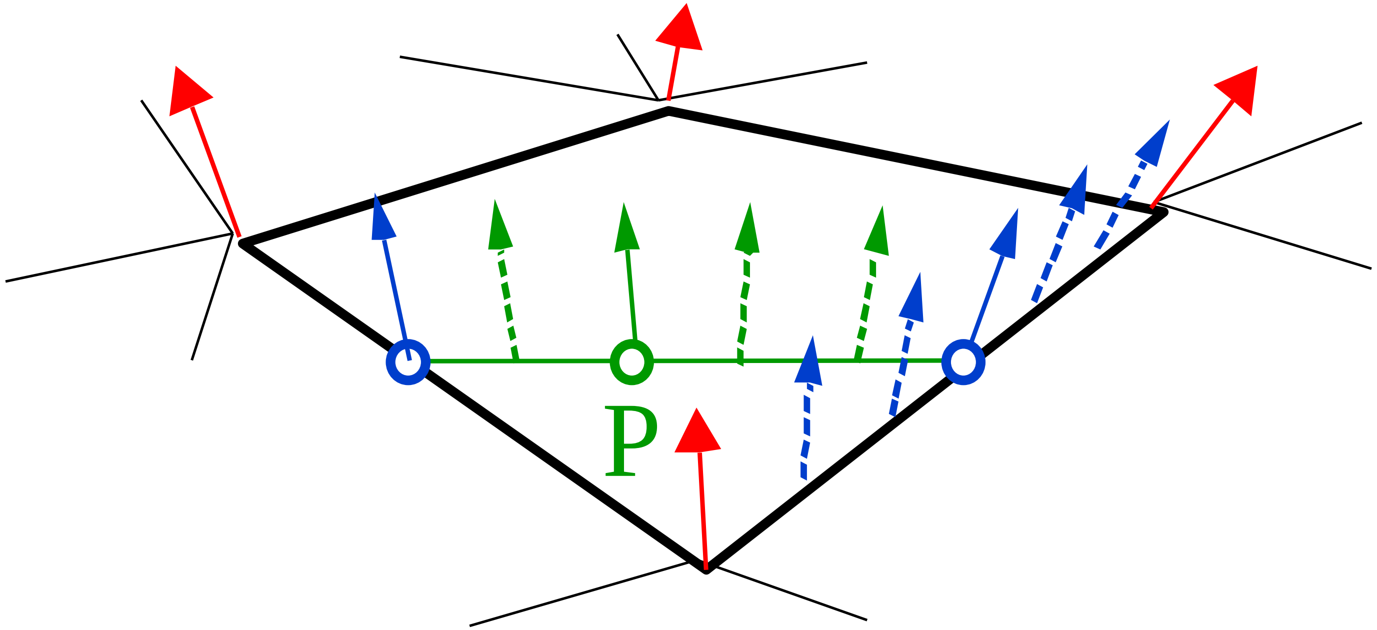
normal-vector interpolation

a. determine average unit normal vector at each polygon vertex

b. linearly interpolate vertex normals

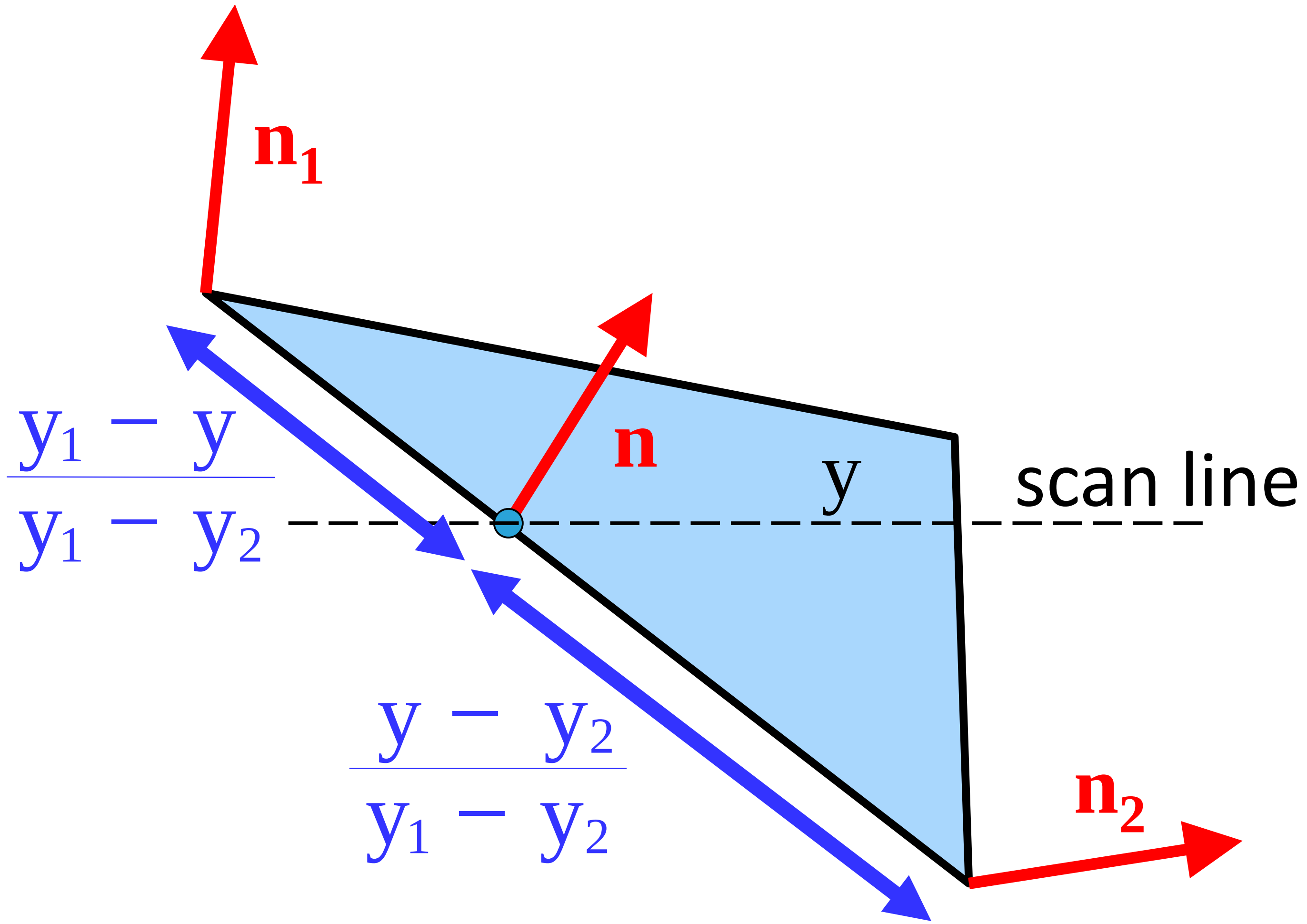
c. apply illumination model along each scan line

Phong Shading Overview



1. normal vectors at corner points
2. interpolate normal vectors along the edges
3. interpolate normal vectors along scanlines
   & calculate shading (intensities) for every pixel

normal-vector interpolation



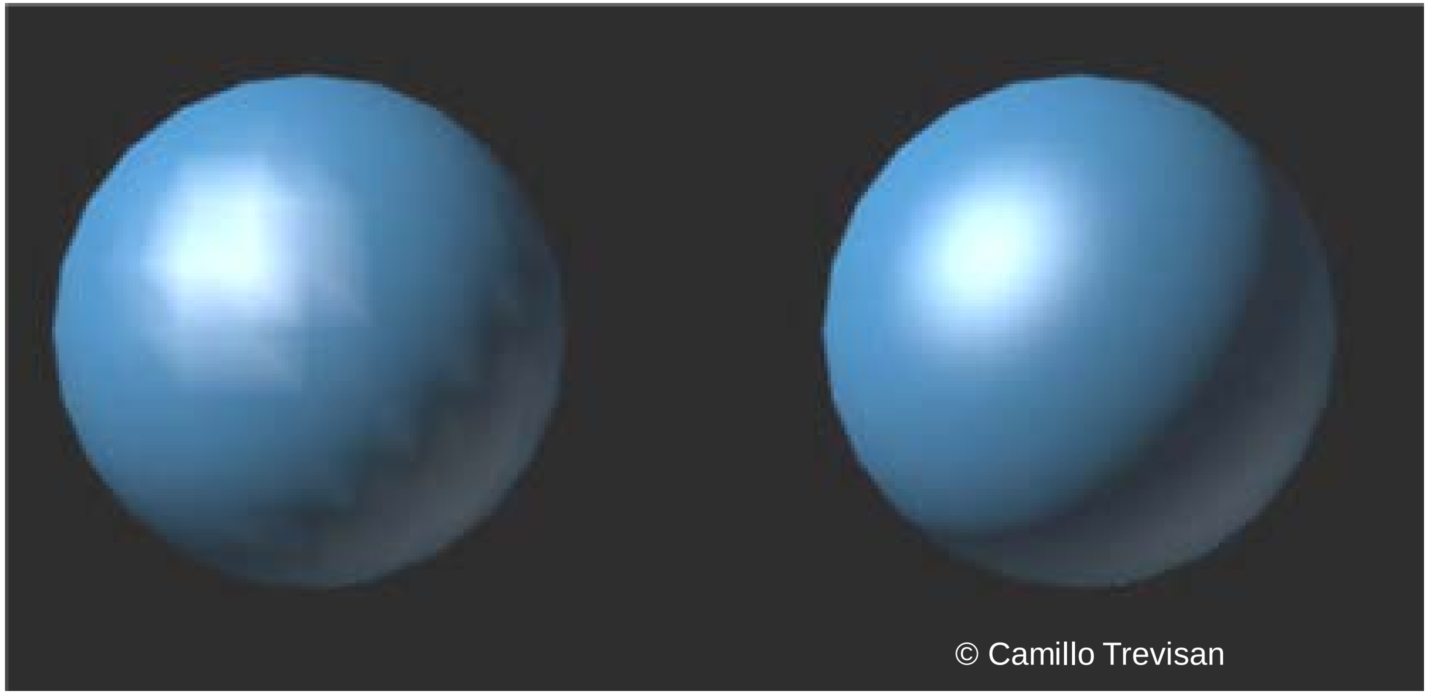
$$\mathbf{n} = \frac{y - y_2}{y_1 - y_2}\mathbf{n}_1 + \frac{y_1 - y}{y_1 - y_2}\mathbf{n}_2$$

Phong Shading

incremental normal vector update along and between scan lines

comparison to Gouraud shading

- better highlights
- less Mach banding
- more costly
- wrong silhouette stays!



Flat/Gouraud/Phong Comparison